



# MAPS Ladder Geometry Norm Vector

Jin Huang(BNL), Anthony Frawley(FSU), Haiwang Yu (NMSU)

# The chain to get the Norm Vector

SvtxCluster → SvtxHit → PHG4CylinderCell\_MAPS → Stave Index

SvtxCluster → Layer → PHG4CylinderGeomContainer → PHG4CylinderGeom\_MAPS

→ Stave Center → Tilt → Norm Vector

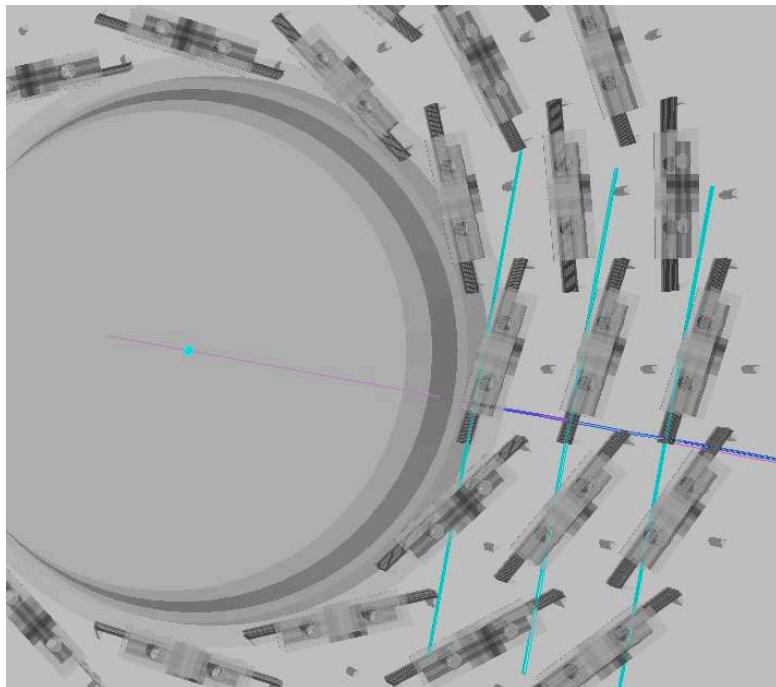
```
TVector3 pos(cluster->get_x(), cluster->get_y(), cluster->get_z());
TVector3 n(cluster->get_x(), cluster->get_y(), 0);

unsigned int layer = cluster->get_layer();
std::cout << "cluster layer: " << layer << std::endl;
if (_detector_type == MAPS_TPC and layer < 3) {

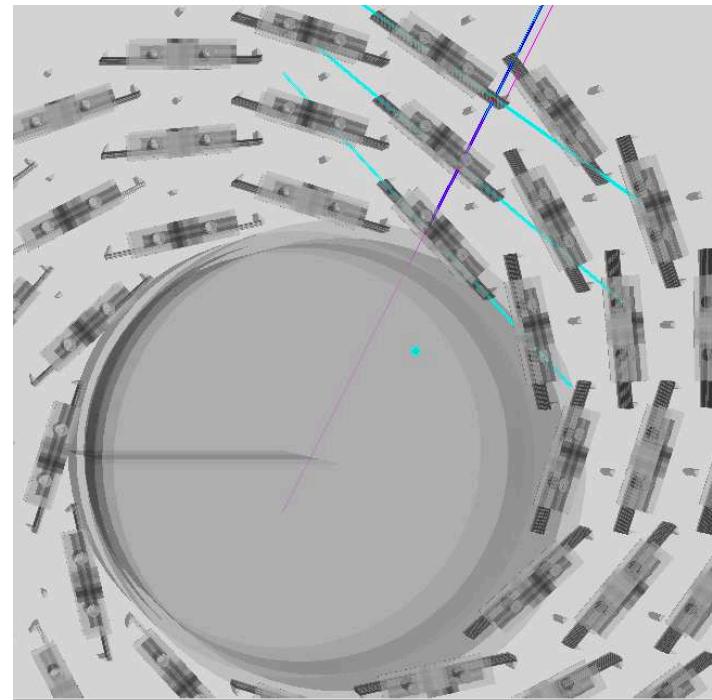
    unsigned int begin_hit_id = *(cluster->begin_hits());
    LogDebug(begin_hit_id);
    SvtxHit* hit = hitsmap->find(begin_hit_id)->second;
    LogDebug(hit->get_cellid());
    PHG4CylinderCell_MAPS* cell = (PHG4CylinderCell_MAPS*) cells->findCylinderCell(hit->get_cellid());
    int stave_index = cell->get_stave_index();
    int half_stave_index = cell->get_half_stave_index();
    int module_index = cell->get_module_index();
    int chip_index = cell->get_chip_index();

    double ladder_location[3] = { 0.0, 0.0, 0.0 };
    PHG4CylinderGeom_MAPS *geom =
        (PHG4CylinderGeom_MAPS*) geom_container->GetLayerGeom(
            layer);
    // returns the center of the sensor in world coordinates - used to get the ladder phi location
    geom->find_sensor_center(stave_index, half_stave_index,
        module_index, chip_index, ladder_location);
    n.Print();
    n.SetXYZ(ladder_location[0],ladder_location[1],0);
    double phitilt = 0.304; // radians, equivalent to 17.4 degrees
    n.RotateZ(phitilt);
    n.Print();
}
```

# Norm Vector

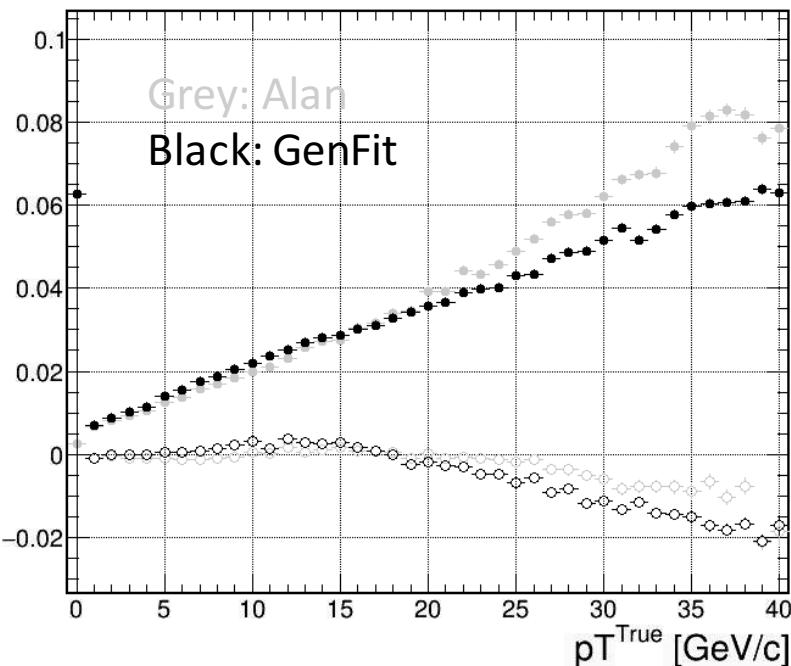


Zoom in



# Momentum resolution, Single pion simulation

pT resolution



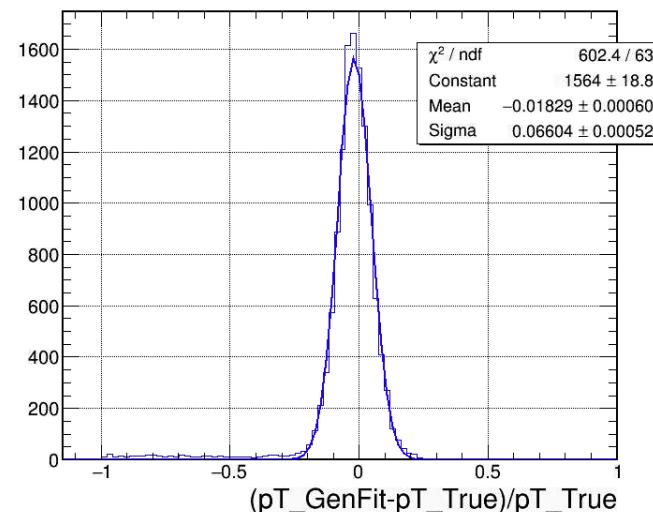
Grey: Alan

Black: GenFit

pT\_Alan has some recalibration:

```
//-----
// Momentum Recalibration
//-----
TF1 *corr = new TF1("corr","1.0/(1+0.00908642+5.91337e-05*x+-1.872
01e-05*x*x+-3.31928e-06*x*x*x+1.03004e-07*x*x*x*x+-1.05111e-09*x*x*x
*x*x)",0.0,40.0);
PHG4SvtxMomentumRecal* recal = new PHG4SvtxMomentumRecal("PHG4Svtx
MomentumRecal",corr);
se->registerSubsystem(recal);
```

(pT\_GenFit-pT\_True)/pT\_True {pT\_True>35}



(pT\_Alan-pT\_True)/pT\_True {pT\_True>35}

